```
 1: /*  SSS_Enhanced.cpp
 2:
 3:     This program solves the side-side-side ( SSS ) problem in trigonometry.
 4:
 5:     More specifically, if the lengths of three sides of a triangle are known,
 6:     this program will solve for the three angles.
 7:
 8:     Units:  The units of input for the three side lengths is actually
 9:     unimportant, so long as all three sides are given in the same units.
10:     This program reports results in both radians and degrees.
11:
12:     Written August 2011 by John Bell, as a sample solution to HW1 for CS 109
13:
14:     This version enhances the basic assignment, by adding a number of
15:     optional enhancements:
16:
17:             1. User input is checked for validity.
18:             2. The sum of the angles is compared to 180 for round-off error,
19:                using both doubles and floats.
20:             3. Heights and areas of triangles are calculated
21:             4. Multiple problems may be solved.
22:
23: */
24:
25: #include <iostream>       // For input and output
26: #include <cmath>          // For sines and cosines, etc.
27: #include <iomanip>        // To format the output nicely.
28:
29: using namespace std;      // For easy use of cin and cout
30:
31: int main( void ) {
32:
33:     // First to declare necessary variables.
34:
35:     double sideA, sideB, sideC;              // 3 sides, any dimensions
36:     double angleA, angleB, angleC;              // in radians
37:     double angleA_deg, angleB_deg, angleC_deg; // in degrees
38:     double numerator, denominator;              // For use in cosine law
39:     double heightA, heightB, heightC;           // Relative to base sideA, etc.
40:     double areaA, areaB, areaC;                 // sideA * heightA / 2.0, etc.
41:     double error;        // 180 - sum( angles ), should be zero
42:     char answer;         // To stop infinite loop.
43:     float fsideA, fsideB, fsideC, fangleA, fangleB, fangleC,
44:           fangleA_deg, fangleB_deg, fangleC_deg, fnumerator, fdenominator,
45:           ferror; // Same as double versions, to compare error of floats.
46:
47:     // Next to explain the program to the user
48:
49:     cerr << "\nWelcome to program SSS_Enhanced.\n\n";
50:     cerr << "Written August 2011 by John Bell, jbell, for CS 109.\n\n";
51:     cerr << "This program will find the three angles of a triangle,\n";
52:     cerr << "given the lengths of the three sides.\n\n";
53:     cerr << "Input can be given in any consistent units.\n";
54:     cerr << "Results are reported in both radians and degrees.\n\n";
55:
56:     cerr << "This enhanced version also calculates heights and areas,\n";
57:     cerr << "and reports observed round-off errors.\n";
```

```
 58:     cerr << "Multiple problems may also be solved.\n\n";
 59:
 60:
 61:     // Now to begin the work
 62:
 63:     while( true ) { // Infinite loop to solve multiple problems.
 64:
 65:         // Now get input from the user.
 66:         // Check for valid input and repeat as needed.
 67:
 68:         while( true ) {
 69:                cerr << "Please enter the length of the longest side: ";
 70:                cin >> sideC;
 71:                if( sideC > 0.0 )
 72:                    break;
 73:                cerr << "Error: " << sideC << " is not greater than zero.\n";
 74:                cerr << "Please try again.\n";
 75:         } // while loop checking sideC
 76:
 77:         while( true ) {
 78:                cerr << "Now please enter the length of the second side: ";
 79:                cin >> sideA;
 80:                if( sideA > 0.0 && sideA <= sideC )
 81:                    break;
 82:                cerr << "Error: " << sideA << " is not between 0.0 and "
 83:                     << sideC << endl;
 84:                cerr << "Please try again.\n";
 85:         } // while loop checking sideA
 86:
 87:         while( true ) {
 88:             cerr << "And finally, please enter the length of the third side: ";
 89:             cin >> sideB;
 90:             if( sideB > sideC - sideA && sideB <= sideC )
 91:                 break;
 92:             cerr << "Error: " << sideB << " is not between " << sideC - sideA
 93:                  << " and " << sideC << endl;
 94:             cerr << "Please try again.\n";
 95:         } // while loop checking sideB
 96:
 97:         // Convert the double inputs into floats
 98:
 99:         fsideA = ( float ) sideA;
100:         fsideB = ( float ) sideB;
101:         fsideC = ( float ) sideC;
102:
103:         // First calculate angle C using the cosine law; Repeat with floats
104:
105:         numerator = sideA * sideA + sideB * sideB - sideC * sideC;
106:         denominator = 2.0 * sideA * sideB;
107:         angleC = acos( numerator / denominator );
108:
109:         fnumerator = fsideA * fsideA + fsideB * fsideB - fsideC * fsideC;
110:         fdenominator = 2.0f * fsideA * fsideB;
111:         fangleC = acos( fnumerator / fdenominator );
112:
113:         // Next calculate angles A and B using the sine law;  Repeat w. floats
114:
```

```
115:         angleA = asin( sideA / sideC * sin( angleC ) );
116:         angleB = asin( sideB / sideC * sin( angleC ) );
117:
118:         fangleA = asin( fsideA / fsideC * sin( fangleC ) );
119:         fangleB = asin( fsideB / fsideC * sin( fangleC ) );
120:
121:         // Next calculates heights perpindicular to each side
122:
123:         heightC = sideB * sin( angleA );
124:         heightA = sideC * sin( angleB );
125:         heightB = sideC * sin( angleA );
126:
127:         // And the areas
128:
129:         areaA = sideA * heightA / 2.0;
130:         areaB = sideB * heightB / 2.0;
131:         areaC = sideC * heightC / 2.0;
132:
133:         // Convert angles from radians to degrees;  Repeat with floats
134:
135:         angleA_deg = angleA * 180.0 / M_PI;  // M_PI defined in <cmath>
136:         angleB_deg = angleB * 180.0 / M_PI;
137:         angleC_deg = angleC * 180.0 / M_PI;
138:
139:         fangleA_deg = fangleA * 180.0f / M_PI;
140:         fangleB_deg = fangleB * 180.0f / M_PI;
141:         fangleC_deg = fangleC * 180.0f / M_PI;
142:
143:         // Calculate round-off errors in degree totals, in double and float
144:
145:         error = 180.0 - ( angleA_deg + angleB_deg + angleC_deg );
146:
147:         ferror = 180.0f - ( fangleA_deg + fangleB_deg + fangleC_deg );
148:
149:         // And finally, echo input and report results.
150:         // Formatting is applied in this version.
151:
152:         cout << "\n\nHere are your results:\n\n";
153:         cout << "     Side        Opposing Angle         Height        Area\n";
154:         cout << "    Length   ( radians ) ( degrees )\n";
155:
156:         cout.unsetf( ios::scientific );
157:         cout.setf( ios::fixed );
158:
159:         cout << setw( 10 ) << setprecision( 4 ) << sideC
160:              << setw( 11 ) << setprecision( 4 ) << angleC
161:              << setw( 13 ) << setprecision( 4 ) << angleC_deg
162:              << setw( 12 ) << setprecision( 4 ) << heightC
163:              << setw( 13 ) << setprecision( 4 ) << areaC << endl;
164:
165:         cout << setw( 10 ) << setprecision( 4 ) << sideA
166:              << setw( 11 ) << setprecision( 4 ) << angleA
167:              << setw( 13 ) << setprecision( 4 ) << angleA_deg
168:              << setw( 12 ) << setprecision( 4 ) << heightA
169:              << setw( 13 ) << setprecision( 4 ) << areaA << endl;
170:
171:         cout << setw( 10 ) << setprecision( 4 ) << sideB
```

```
172:                << setw( 11 ) << setprecision( 4 ) << angleB
173:                << setw( 13 ) << setprecision( 4 ) << angleB_deg
174:                << setw( 12 ) << setprecision( 4 ) << heightB
175:                << setw( 13 ) << setprecision( 4 ) << areaB << endl << endl;
176:
177:          cout.unsetf( ios::fixed );
178:          cout.setf( ios::scientific );
179:          cout << "The error in the sum of the angles is\n\t"
180:                << error << " using doubles, and\n\t "
181:                << ferror << " using floats." << endl << endl;
182:
183:          // Now to check and see if they want to solve any more problems
184:
185:          cerr << "Would you like to solve another problem?  Answer Y or N: ";
186:          cin >> answer;
187:          if( answer != 'Y' && answer != 'y' )
188:              break;
189:
190:      } // End of the infinite while loop
191:
192:      // Tha tha tha that's all folks !
193:
194:      system( "PAUSE" );    // Only needed for Dev C++
195:
196:      return 0;
197:
198: } // main
199:
```